2020

IDENTIFYING OPIOID WITHDRAWAL USING WEARABLE BIOSENSORS

Ethan Kulman
University of Rhode Island, ethan_kulman@my.uri.edu

IDENTIFYING OPIOID WITHDRAWAL USING WEARABLE BIOSENSORS

BY

ETHAN KULMAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

UNIVERSITY OF RHODE ISLAND

2020

MASTER OF SCIENCE THESIS

OF

ETHAN KULMAN

APPROVED:

Thesis Committee:

Major Professor Krishna Kumar Venkatasubramanian

Marco Alvarez

Kunal Mankodiya

Brenton DeBoef

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2020

ABSTRACT

Wearable biosensors can be used to monitor opioid use, a problem of dire societal consequence given the current opioid epidemic in the US. Such surveillance can prompt interventions that promote behavioral change. Prior work has focused on the use of wearable biosensor data to detect opioid use. In this work, we present a method that uses machine learning to identify opioid withdrawal using data collected with a wearable biosensor. Our method involves developing a set of machine-learning classifiers, and then evaluating those classifiers using unseen test data. An analysis of the best performing model produced a receiver operating characteristic (ROC) area under the curve (AUC) of 0.9997 using completely unseen test data. Further, the model is able to detect withdrawal with just one minute of biosensor data. These results show the viability of using machine learning for opioid withdrawal detection. To our knowledge, the proposed method for identifying opioid withdrawal in OUD patients is the first of its kind.

ACKNOWLEDGMENTS

I would first like to thank Dr. Kunal Mankodiya, Dr. Marco Alvarez, and Dr. Krishna Kumar Venkatasubramanian for being on my thesis commitee. I would like to thank all of my family and friends who supported me through my graduate education. I want to especially thank Dr. Krishna Kumar Venkatasubramanian for advising me through the thesis process. I want to thank Dr. Stephanie Carreiro, and Brittany Chapman from the University of Massachusetts Medical School for helping provide me with the dataset for my analysis and clinical advice related to my research. I would also like to thank Dr. Edmund Lamagna and Dr. Marco Alvarez for encouraging me to pursue a graduate education, and challenging me in their classes with rigorous coursework. And I would like to thank Dr. Thomas Sproul for his support over the years, and for being the External Chair member on my thesis committee.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1.1 Introduction and Motivations

The Center For Disease Control (CDC) has reported that of the 70,000 people who died from a drug overdose in the United States during 2017, 68% of those deaths involved opioids [1]. The treatment process for individuals with *opioid use disorder (OUD)* involves detoxification (*aka* detox), often with medication assisted treatment (MAT) using drugs such as methadone or buprenorphine [2]. During the detoxification period, OUD subjects can experience *opioid withdrawal symptoms* for up to 7 days after their last drug use. Symptoms of opioid withdrawal include nausea, vomiting, diarrhea, and severe diffuse body pain [3]. These symptoms are often so severe that they have been found to increase the risk of relapse and overdose death [3]. Some studies have even shown that up to 70% of OUD subjects relapse after completing opioid detoxification due to the withdrawal they experience. [4]. The process of opioid detoxification is complicated and difficult for both healthcare practitioners and subjects.

Up until the past few decades, the evaluation of patient health and wellbeing was limited to when a patient visited their healthcare provider [5]. More recently, the improvements in commercially available *wearable biosensors* have given healthcare providers the capability to monitor various aspects of the physiological state of their patients health remotely [5]. These analytical devices can be worn at all times by patients, and can collect and transmit key indicators of patient physiology in real time. Wearable biosensors have already been shown to have potential for detecting and managing opioid use in real-time [6] [7] [8]. Undergoing the detoxification process, and experiencing opioid withdrawal can be difficult for OUD patients. Improving the clinician's ability to monitor subjects during withdrawal

(e.g. while in a detoxification program) would help clinicians personalize treatment options for relapse prevention. Personalized treatment for opioid withdrawal has the potential to improve treatment success and ultimately save lives.

In this work, *we present a method that uses machine learning to identify opioid withdrawal using data collected with a wearable biosensor.* To develop and evaluate our approach for detecting opioid withdrawal using biosensors, we rely on using biosensor data collected from overdosing subjects in a hospital emergency department (ED). We used an Empatica E4 wrist-mounted biosensor (Empatica, Milan, Italy) for our data collection. Data was collected from 16 subjects who presented to a single ED for medical care following an opioid overdose. The subjects were in various states of recovery subsequent to an administration of naloxone [1]. Our subjects were real medical patients suffering from OUD, and all data was gathered in a way that prioritized subject care and wellness over research goals with approval from our Institutional Review Board (IRB).

In order to detect withdrawal, we use standard machine learning techniques to develop classifiers that capture the uniqueness of the physiological measurements collected by the Empatica E4 during withdrawal. The physiological measurements collected are blood volume pulse, electrodermal activity, skin temperature, and movement (accelerometry). During their stay in the ED, the subjects were evaluated by clinicians every 30 minutes to an hour. At each of these evaluations the subjects were assessed to be in one of three *states* – withdrawal, intoxicated, or neutral. We decided to use a 20 minute interval surrounding the time when the physician assessed the physiological state of a subject for training and testing our models, as we had confidence in the ground-truth of a subject's state during that time. The classifiers developed were general models essentially able to distinguish

[1] Naloxone is an antidote that is given to someone who is overdosing on opioids. It immediately reverses the effect of opioids by competitively binding to the opioid receptors in the body

the withdrawal state from all other states.

In total, our dataset had data collected from 16 different OUD subjects. Six of the 16 subjects had data in the withdrawal state. Two of these six subjects had neutral data as well, the other four had data exclusively in the withdrawal state (based on the clinician's assessment). The remaining 10 other OUD subjects used in this study had data assessed in either the neutral or the intoxicated states or both. This means our dataset has many more examples in the neutral and intoxicated states compared to the withdrawal state. This class imbalance had to be addressed during the development of our models.

The analysis presented in this work demonstrates the **viability** of our method. Upon training our models, and compensating for the class imbalance, we were able to achieve almost perfect results using our test data. The best performing model (Random Forest) during testing had a receiver operating characteristic (ROC) area under the curve (AUC) of 0.9997. Our test data was *completely unseen data* (by our models during training) including both withdrawal and non-withdrawal states. Further, the model is able to detect withdrawal with just one minute of biosensor data. To the best of our knowledge, this is the first work related to using machine learning to identify opioid withdrawal of any kind.

List of References

[1] "CDC's efforts to prevent opioid overdoses and other opioid-related harms," Nov 2019. [Online]. Available: https://www.cdc.gov/opioids/framework/index.html

[2] G. L. Bailey, D. S. Herman, and M. D. Stein, "Perceived relapse risk and desire for medication assisted treatment among persons seeking inpatient opiate detoxification," *Journal of substance abuse treatment*, vol. 45, no. 3, pp. 302–305, 2013.

[3] J. K. Nuamah, F. Sasangohar, M. Erranguntla, and R. K. Mehta, "The past, present and future of opioid withdrawal assessment: a scoping review of scales

and technologies," *BMC medical informatics and decision making*, vol. 19, no. 1, p. 113, 2019.

[4] H. Chalana, T. Kundal, V. Gupta, and A. S. Malhari, "Predictors of relapse after inpatient opioid detoxification during 1-year follow-up," *Journal of addiction*, vol. 2016, 2016.

[5] K. Guk, G. Han, J. Lim, K. Jeong, T. Kang, E.-K. Lim, and J. Jung, "Evolution of wearable devices with real-time disease monitoring for personalized healthcare," *Nanomaterials*, vol. 9, no. 6, p. 813, 2019.

[6] M. S. Mahmud, H. Fang, H. Wang, S. Carreiro, and E. Boyer, "Automatic detection of opioid intake using wearable biosensor," in *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018, pp. 784–788.

[7] S. Carreiro, K. Wittbold, P. Indic, H. Fang, J. Zhang, and E. W. Boyer, "Wearable biosensors to detect physiologic change during opioid use," *Journal of medical toxicology*, vol. 12, no. 3, pp. 255–262, 2016.

[8] R. Singh, B. Lewis, B. Chapman, S. Carreiro, and K. Venkatasubramanian, "A machine learning-based approach for collaborative non-adherence detection during opioid abuse surveillance using a wearable biosensor," in *Biomedical engineering systems and technologies, international joint conference, BIOSTEC... revised selected papers. BIOSTEC (Conference)*, vol. 5. NIH Public Access, 2019, p. 310.

CHAPTER 2

Literature Review

2.1 Related Work

As previously mentioned, we do not know of any work that has been done related to using wearable biosensors to detect opioid withdrawal. The majority of research involving identifying opioid withdrawal is related to the development of clinical tools whose purpose is to assess a patient for withdrawal symptoms. The common form these clinical tools come in are surveys or scales [1]. One commonly used assessment tool is the the Clinical Opiate Withdrawal Scale (COWS). This scale considers a number of different physiological symptoms to help medical staff identify to what extent a patient is experiencing opioid withdrawal [2]. These opioid withdrawal scales have limitations since signs and symptoms may go unrecorded when clinicians are not observing a patient, and they require patients to self report certain symptoms [1]. There has also been previous pharmacological research that conceptualizes opioid withdrawal as a stressor [3].

There are many studies that have used machine learning to identify stress. Data collected from wearable biosensors has been used to successfully build both personalized and general models for detecting stress. [4] [5].

Limited work has been done related to stress detection in Substance Use Disorder (SUD) patients [6], but no such work has been done related to OUD patients. However, since it has been shown that opioid withdrawal symptoms can be conceptualized as being similar to those of stress [3], we will leverage some of the features of stress detection in this work. Specifically, there are two works related to using wearable biosensors to detect stress which heavily influenced the methodology for this research. The synopsis for these two studies are detailed below.

The first work titled "Real-Time Monitoring of Passenger's Psychological

Stress" by Vila et. al. [4] seeked to develop a personalized stress detection model using data collected from a wearable biosensor. In this study, the Empatica E4 was worn by the chosen subject during business travel to collect training data. During the collection of data, the subject was asked to rate their stress during different activities on a scale from 1 to 10. The training data was then labeled with the self-reported stress levels. The sensor during this time collected data for blood volume pulse (BVP), electrodermal activity (EDA), and 3 axis acceleration. This training dataset was then broken into non-overlapping 60 second windows, where each window had 24 different features extracted from it. The 24 training features extracted from a particular 60 second window represented one training instance. The training data was then used to develop a linear regression model to predict a label of stressed, or not-stressed. This personalized linear regression model for detecting stress was deployed on a mobile application to test the ability of the model to label stress in real-time. The participant then went on travel for business again to collect test data, only this time the Empatica E4 was paired with the already trained personalized stress detection model on the participants phone. This trained personalized stress detection model labeled the data collected in real-time, and at the same time the participant also labeled their data manually (as stress or not-stressed). The real-time prediction of the algorithm was compared with the participants labeling of their stress levels during this test phase. The model was successful at distinguishing between stress and non-stress, and correctly labeled large chunks of time reported as least stressful (1/10 stress level) 96.5 % $\pm$ 3.2. It was stated in this paper that the features extracted for the model were not supposed to be the optimal feature set for detecting stress, but a baseline of features which can be used to detect stress in real-time. [4]

In the second work, "Continuous Stress Detection Using Wearable Sensors in

Real Life: A Programming Contest Case Study" [5], 21 students were enrolled during a programming camp to build a general model for classifying stress. Each participant was assigned to wear a smart watch to collect their biometric data (2 Samsung Gear S1, 10 Samsung Gear S2, 4 Samsung Gear S3 smartwatches and 4 Empatica E4 wristbands). Three different activities were performed during the camp: free time, lecture, and competition. Each activity was given a pre-defined stress level by the researchers (free time: low, lecture: medium, competition: high). They obtained a subjective stress level by asking participants during their activities what their stress level was on a 0-100 scale with increments of 5 (0-30 is low, 35-75 is medium, 80-100 is high). They collected data for blood volume pulse (BVP), electrodermal activity (EDA), and 3 axis acceleration. This data was used to train and test 6 different models (PCA + LDA, PCA + SVM (radial), kNN, Logistic Regression, Random Forest, Multi-layered perceptron). The models built using the pre-defined stress level, and the subjective stress level were both successful at identifying stress. The top performing models were the Random Forest and Multi-layered perceptron. The best general model for identifying stress had an accuracy of 88.20%, while the best person-specific model had an accuracy of 97.92%.

Wearable biosensors have also been used in opioid research for automatic detection of opioid intake [7] [8], and detecting recurrent opioid toxicity in patients after being administered naloxone [9]. The non-adherence of opioid abuse disorder patients wearing a biosensor has also been recently investigated [10]. None of these previous studies looked to use wearable biosensors to identify opioid withdrawal. Therefore, there is a need to explore using wearable biosensors to detect opioid withdrawal.

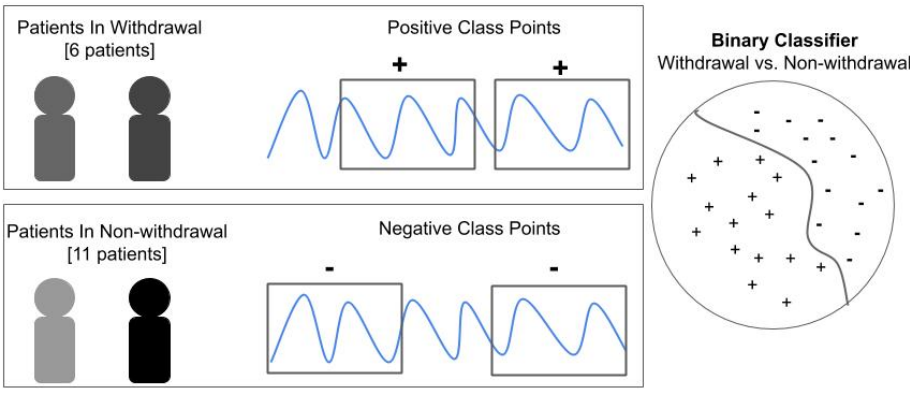

Figure 1: Overview of problem statement for developing a classifier to detect opioid withdrawal.

2.2 Problem Statement

The goal of this paper is to explore the use of machine learning for identifying opioid withdrawal. The idea is to build a model that learns to differentiate physiological data (collected from wearable biosensors) assessed to be in the withdrawal state, from data assessed in either the neutral or intoxicated state (i.e., non-withdrawal state). Once developed, this model will be able to assess whether or not a never before seen snippet of data has come from an OUD patient in the withdrawal state. As shown in Figure 1, we aim to build a binary classifier that can distinguish between biosensor data emanating from a person in withdrawal versus a person not in withdrawal.

List of References

[1] J. K. Nuamah, F. Sasangohar, M. Erranguntla, and R. K. Mehta, "The past, present and future of opioid withdrawal assessment: a scoping review of scales and technologies," *BMC medical informatics and decision making*, vol. 19, no. 1, p. 113, 2019.

[2] D. R. Wesson and W. Ling, "The clinical opiate withdrawal scale (cows)," *Journal of psychoactive drugs*, vol. 35, no. 2, pp. 253–259, 2003.

[3] E. H. Chartoff and W. A. Carlezon Jr, "Drug withdrawal conceptualized as a stressor," *Behavioural pharmacology*, vol. 25, p. 473, 2014.

[4] G. Vila, C. Godin, O. Sakri, E. Labyt, A. Vidal, S. Charbonnier, S. Ollander, and A. Campagne, "Real-time monitoring of passenger's psychological stress," *Future Internet*, vol. 11, no. 5, p. 102, 2019.

[5] Y. S. Can, N. Chalabianloo, D. Ekiz, and C. Ersoy, "Continuous stress detection using wearable sensors in real life: Algorithmic programming contest case study," *Sensors*, vol. 19, no. 8, p. 1849, 2019.

[6] S. Carreiro, K. K. Chintha, S. Shrestha, B. Chapman, D. Smelson, and P. Indic, "Wearable sensor-based detection of stress and craving in patients during treatment for substance use disorder: A mixed methods pilot study," *Drug and Alcohol Dependence*, p. 107929, 2020.

[7] M. S. Mahmud, H. Fang, H. Wang, S. Carreiro, and E. Boyer, "Automatic detection of opioid intake using wearable biosensor," in *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018, pp. 784–788.

[8] S. Carreiro, D. Smelson, M. Ranney, K. J. Horvath, R. W. Picard, E. D. Boudreaux, R. Hayes, and E. W. Boyer, "Real-time mobile detection of drug use with wearable biosensors: a pilot study," *Journal of Medical Toxicology*, vol. 11, no. 1, pp. 73–79, 2015.

[9] K. K. Chintha, P. Indic, B. Chapman, E. W. Boyer, and S. Carreiro, "Wearable biosensors to evaluate recurrent opioid toxicity after naloxone administration: a hilbert transform approach," in *Proceedings of the... Annual Hawaii International Conference on System Sciences. Annual Hawaii International Conference on System Sciences*, vol. 2018. NIH Public Access, 2018, p. 3247.

[10] R. Singh, B. Lewis, B. Chapman, S. Carreiro, and K. Venkatasubramanian, "A machine learning-based approach for collaborative non-adherence detection during opioid abuse surveillance using a wearable biosensor," in *Biomedical engineering systems and technologies, international joint conference, BIOSTEC... revised selected papers. BIOSTEC (Conference)*, vol. 5. NIH Public Access, 2019, p. 310.

CHAPTER 3

Methodology

In this chapter we first describe the collection, and cleaning of our dataset. Next, we will describe the feature extraction process, and an overview of the training and testing procedures.

3.1 Data Collection

The dataset used in this study was collected from individuals (patients) admitted to the emergency department (ED) who received naloxone after experiencing a potential opioid overdose. Upon obtaining informed consent, research staff placed an Empatica E4 wearable biosensor on the patients non-dominant wrist in order to collect their biometric data. The E4 collects four different types of data from the body.

One of the standard tools used by clinicians to assess whether a patient has opioid withdrawal symptoms is the Clinical Opiate Withdrawal Scale (COWS) [1]. The COWS considers, among other biometrics, a subjects heart rate, perspiration, and acute movements. Given that the Empatica E4 collects some of the exact data types used in the COWS, we used these same biometrics in our analysis. Specifically, in our work we used *blood volume pulse (BVP) (sampled at 64 Hz) data, electrodermal activity (EDA) (sampled at 4 Hz) data, skin temperature (sampled at 4 Hz) data, and triaxial accelerometer (sampled at 32 Hz) data.* While the COWS doesn't consider a patient's skin temperature in its assessment for opioid withdrawal symptoms, there are several other opiate withdrawal scales that do [2].

Along with the data collected by the E4, the physiological state of a patient was assessed and recorded by a board-certified emergency physician and medical toxicologist in the ED. The physiological state of a patient was classified as one

Table 1: Demographics of our dataset

Gender	**Count**	**Avg. Age (std)**
Male	13	34.85 ± 9.89
Female	3	38.33 ± 2.05

of three states based on clinician assessment: neutral, intoxicated, or withdrawal. The neutral state refers to a patient not being in opioid intoxication or opioid withdrawal. The intoxicated state refers to a state with signs and symptoms consistent with opioid intoxication. The withdrawal state refers to a state with signs and symptoms of opioid withdrawal. The assessment of a subjects physiological state took place every 30 minutes to an hour. Subjects assessed in the intoxicated or withdrawal states were generally laying on a hospital stretcher due to incapacity or discomfort, respectively. The neutral state assessments were often done while the participant was performing a variety of activities such as walking, talking, eating, etc. A patient may have been assessed in variety of different states during their enrollment in the study. It was not uncommon for subjects to transition from one state to another during the course of the study: for example to be neutral on one assessment, then to be in a withdrawal state on a second assessment 30 minutes later. The clinicians assessment of the state of the patient provides us with the ground-truth needed to build our models. Overall, we used data from 16 subjects in this study. The demographics of the patient population can be found in Table 1.

3.2 Data Cleaning

All of the data used in this study was visualized graphically to verify the quality of the data. All datatypes collected by the Empatica E4 that were used in this study are described in detail in Section 3.1. Visualizing the data graphically revealed areas of problematic data in both the electrodermal activity (EDA) and

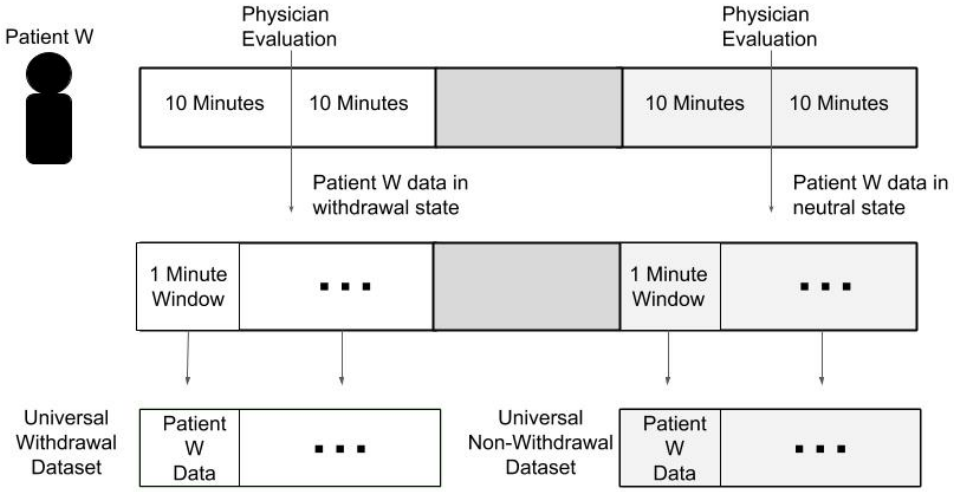

Figure 2: Example of data collection process.

blood volume pulse (BVP) data for some of the patients. These problematic areas or artifacts can be characterized by spikes in the data which are due to noise. Artifacts can appear in both BVP and EDA data due to movement of the biosensor on the skins surface [3] [4]. In order to mitigate the effect of the artifacts present in the BVP and EDA data, a data cleaning process was undertaken for these two data types. The process for cleaning the EDA, and BVP data is detailed in the next few paragraphs.

Specifically, we used filtering techniques to clean the EDA and BVP data. In order to mitigate noise in the EDA data, two low-pass Butterworth filters were applied to the data. The first low-pass Butterworth filter had a cutoff of 0.2 Hz, and the second low-pass Butterworth filter had a cutoff of 0.05 Hz. The use of this technique for the purpose of noise reduction in EDA data has been shown to be successful in previous research [5]. The purpose of applying these two low-pass Butterworth filters is to first use the 0.2 Hz low-pass filter to rid the data of drastic peaks caused by noise, and then apply the 0.05 Hz filter to smooth out

the remaining signal. [5]. Figure 3 shows an example of how the two-low pass Butterworth filters applied to EDA data helps mitigate motion artifacts.

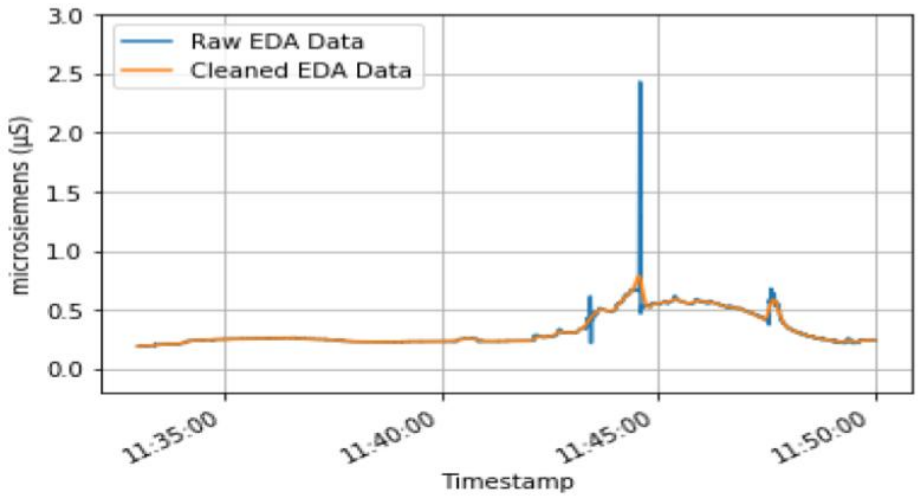

Figure 3: Example of two low-pass Butterworth filters being used to clean a patients EDA data.

The visual inspection of each patients BVP data revealed that noise was spread throughout the signal in every patients data. Noise artifacts in BVP data is a well documented problem, and many different studies have proposed solutions for it [3]. A band-pass filter was the chosen technique to limit the impact of noise present in the BVP data. This band-pass filter had a high-pass cutoff of 0.6 Hz, and a low-pass cut off of 3.33 Hz. These lower and upper frequencies are used to limit the possible heart rates that could appear in this data to a range of 40-200 beats per minute (BPM). This heart rate range accounts for both the upper and low extremes of heart rates that could occur for an individual [6]. Figure 4 shows an example of how the band-pass filter applied to BVP data helped mitigate motion artifacts.

The accelerometer data did not undergo any data cleaning in order to maintain

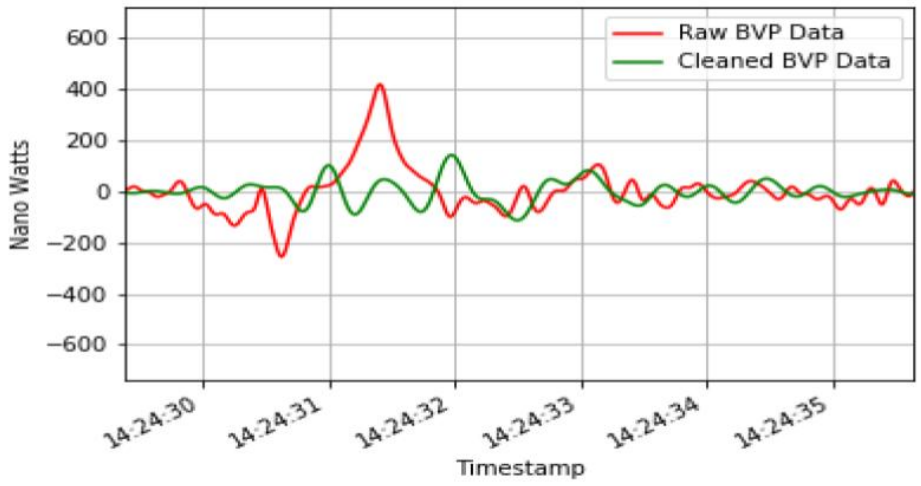

Figure 4: Example of band-pass filter being used to clean a patients BVP data.

acute movements that may be related to opioid withdrawal symptoms. No data cleaning was performed on the skin temperature data, however an inspection of the skin temperature data revealed that some skin temperature readings were far lower than the patients normal skin temperature reading. The sections of data containing abnormally low skin temperature readings were not included in our analysis.

3.3 Dataset Windowing

After cleaning each patients data, it then needs to be discretized. In order to discretize the time series data collected with the Empatica E4 wearable biosensor, a sliding window technique was used. The sliding window technique is a data processing method used to break up continuous data into time windows of a fixed size [7]. The size chosen for a fixed time window is based off of how much time is needed to make a judgement about a particular signal. These fixed sized windows can be non-overlapping, or overlap each-other in order to obtain finer details about

how a signal is changing over time. A non-overlapping window size of one minute was chosen to break up each patients data. A one minute non-overlapping window size was used because this window-size has been used in previous research related to classifying stress using machine learning [5]. As previously mentioned, opioid withdrawal can be conceptualized as being similar to stress [8].

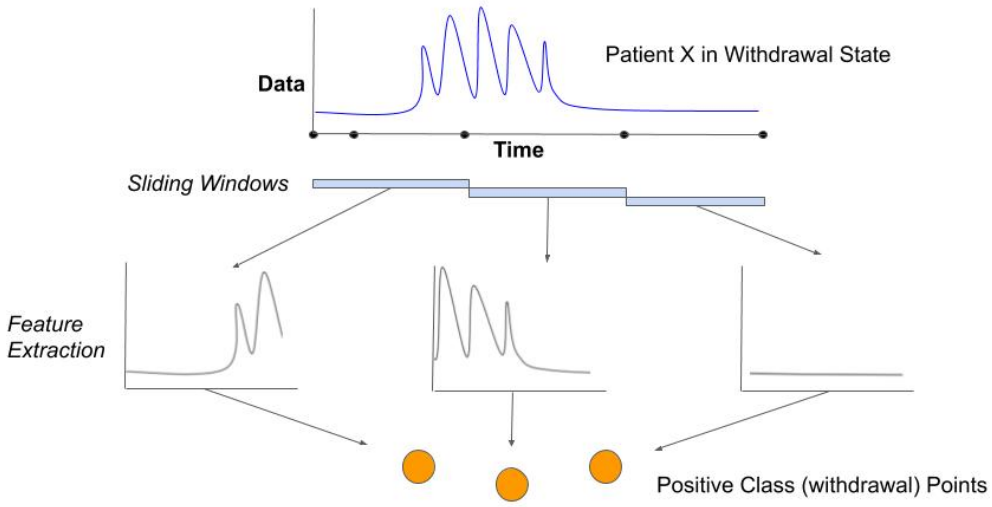

Figure 5: Example of data curation process applied to withdrawal state data.

The one minute segments that were assessed to be in the withdrawal state are placed into one dataset (the universal withdrawal set), and the one minute segments that were assessed to be in the intoxicated or neutral state were placed into a separate dataset (the universal non-withdrawal set) (see Figure 5). Once all patient data was placed into either the universal withdrawal or universal non-withdrawal dataset, the feature extraction process was performed.

3.3.1 Detailed Breakdown of Data Windows

Prior to discussing the feature extraction process, we first provide a detailed breakdown of the one minute data windows used in this analysis. The break down

of how many one minute data windows a subject had in each state, as well as the grand totals is shown in Table 2. Each data window here equates to one class points after the feature extraction process is performed. It is clear that there are far fewer withdrawal state windows compared to the combined total of neutral state and intoxicated state windows. We also show here that many subjects had data in more than one state, and that two of our subjects with withdrawal state data also had data in the neutral state. As previously mentioned in the Section 3.2 (Data Cleaning), certain patients had noisy or abnormal data which was excluded from our analysis. This is why some patients have far fewer one minute data windows compared to others (e.g. Subject N0025 had only 9 minutes of usable neutral state data).

Table 2: Number of data windows extracted from each state for a subject

Subject	**Total Windows**	**Neutral**	**Intoxicated**	**Withdrawal**
N0002	18	18	0	0
N0004	11	0	11	0
N0008	17	0	17	0
N0013	47	19	0	28
N0015	19	19	0	0
N0017	64	7	57	0
N0022	28	0	28	0
N0023	30	19	11	0
N0024	25	25	0	0
N0025	9	9	0	0
N0026	35	0	0	35
N0027	17	17	0	0
N0029	34	19	0	15
N0032	19	0	0	19
N0033	19	0	0	19
N0045	27	0	0	27
Grand Total	**419**	**152**	**124**	**143**

3.4 Feature Extraction

Once all of the data has been broken up into one minute segments and placed into either the universal withdrawal or universal non-withdrawal datasets, we can then generate feature vectors from each one minute of data. *The features extracted for this analysis are inspired by two of the works detailed in Section 2.1 (Related Work) which used wearable biosensors to classify stress* [5] [9].

There are a total of 66 features extracted from each one minute window during feature extraction. These 66 features form a feature-vector that is then labeled as belonging to a positive or negative class. The *positive class* feature-vectors are derived from data collected in the withdrawal state and *negative class* feature-vectors are derived from data collected in the intoxicated or neutral states. The positive and negative class points are detailed further in Section 3.5.

The features extracted from the EDA, BVP, skin temperature, and triaxial accelerometer data are described in Table 3.

In total, there are 11 features extracted from the EDA data, 14 features extracted from the BVP data, 13 different features extracted from each axes (x, y, z) of the triaxial accelerometer data (39 in total), and 2 features extracted from the skin temperature data.

Separability of Features

Prior to detailing how we used these features to train our models, we will provide an intuition for why these features may help a model to distinguish between the withdrawal and non-withdrawal states. We do this by plotting a pair of features for each class point in our analysis: one feature on the x-axis, and one feature on the y-axis.

In Figure 6, we show a pair of features that highlight the separability of the positive and negative class points. Here, we show the mean inter-beat interval

Table 3: Features extracted from each datatype collected by the Empatica E4

Datatype	**Features Extracted**
EDA	mean, mean derivative, standard deviation, number of peaks, mean prominence, mean width between peaks, dot product of the peak width and prominence, number of strong peaks, 20th percentile, quartile, 80th percentile
BVP	mean inter-beat interval (IBI), IBI standard deviation, root mean square, total power of IBI, low frequency power of IBI, high frequency power of IBI, normalized low frequency power of IBI, normalized high frequency power of IBI, low frequency power to high frequency power ratio of IBI, number of peaks, mean amplitude, standard deviation of amplitude, square average, percent of IBI greater than 50 milliseconds.
Triaxial Accelerometer (x, y, z)	total power, mean absolute difference of the norm, mean derivative, mean, median, skew, variance, standard deviation, maximum, minimum, interquartile range, zero crossing rate, kurtosis
Skin Temperature	mean, mean derivative

feature plotted against the mean EDA feature for each class point. The mean EDA is a measurement of the average conductivity level of the skin [10]. The higher the conductivity level of the skin or EDA, the more a person is sweating [10]. The mean inter-beat interval is the average amount of time, in milliseconds, between heart beats [11]. Although there is quite a bit of overlap between withdrawal class and non-withdrawal class samples at low values of the mean EDA (between 0 and 0.25 microsiemens), the majority of non-withdrawal points have a lower EDA, and higher inter-beat interval. This is what one would expect to see. When a person is not in withdrawal they are *less* sweaty and their heart is not beating very fast. The opposite is however true, generally speaking, when one is in withdrawal.

Specifically, about 40% of non-withdrawal class points have an EDA below 0.15 microsiemens, compared to only 13% of withdrawal class points. At the same time, over 70% of withdrawal class points have an inter-beat interval below 35 milliseconds, compared to only 50% of the non-withdrawal points. A low inter-beat interval (or faster heart rate), and a higher average EDA (sweatiness) are exactly what scales like the COWS expect to find in a patient experiencing opioid withdrawal [1].

3.5 Training And Testing Overview

Once we have the dataset and know which features to extract, the next step is to build the opioid withdrawal detection model. Our detection model uses a machine learning-based classifier to address our principal question.

Our classifier attempts to learn the uniqueness of the EDA, accelerometer, temperature, and BVP data (collected using the wearable biosensor) for the OUD subjects withdrawal state. Once the model is built, any newly received EDA, accelerometer, temperature, and BVP data snippet which matches the models understanding of the withdrawal state will be classified as such. Our detection

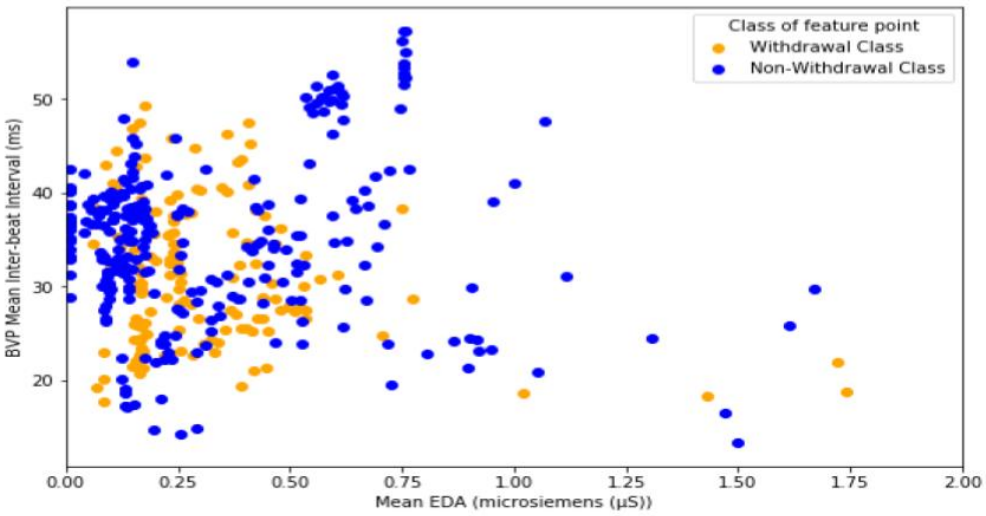

Figure 6: Comparison of the mean inter-beat interval and EDA mean values for each class point.

approach has two phases: the training phase, and the testing phase.

3.5.1 Training Phase

The goal of the training phase is to develop a machine-learning model (specifically, a binary classifier) for identifying opioid withdrawal, where our model needs to be able to recognize the withdrawal state from a variety of non-withdrawal states. In order to do this, we must first label the subjects data into one of two different classes. **(1) Positive Class:** The positive class consists of all 66-point feature vectors from the six different subjects whose data were assessed to be in the withdrawal state. **(2) Negative Class:** The negative class consists of all 66-point feature vectors from the 12 different subjects whose data were assessed to be in the non-withdrawal state. In our study, the non-withdrawal class refers to subject data assessed in the neutral and intoxicated states. These two physiological states are lumped together into the non-withdrawal state because the primary goal of our analysis is to evaluate how well a machine-learning classifier can dis-

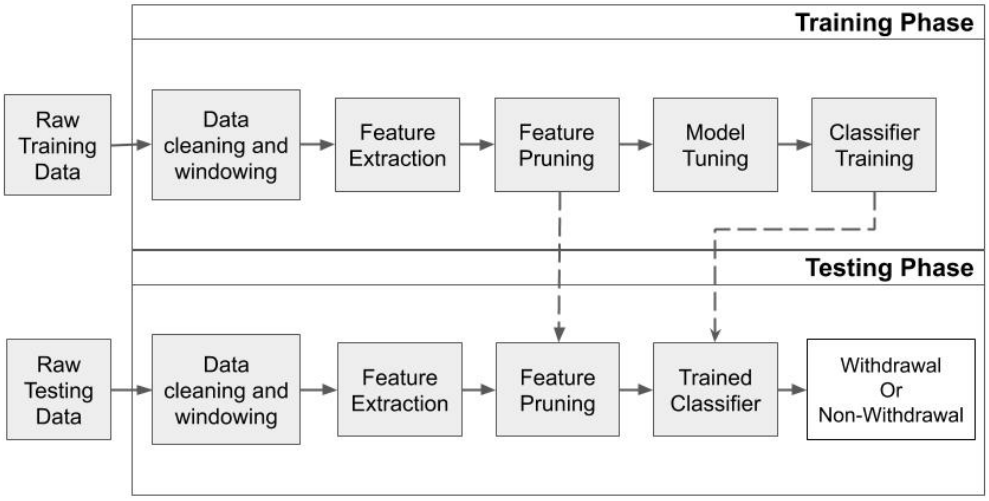

Figure 7: Overview of the training and testing methodology.

tinguish the withdrawal state from other physiological states found in our subjects population.

3.5.2 Testing Phase

Once the machine learning model is trained, it is now able to classify whether a 66-point feature vector, derived from a *never before seen* one minute snippet of EDA, accelerometer, temperature, and BVP measurements, belongs to the withdrawal or non-withdrawal state. Since we are performing binary classification (withdrawal vs. non-withdrawal), our classifier typically returns a confidence value from 0 to 1, with 1 indicating that the model has full confidence that the unseen snippet belongs withdrawal state, and with 0 indicating full confidence that the point belongs to the non-withdrawal state. We are then able to decide whether to accept or reject that value depending on whether or not it meets a chosen threshold between 0 and 1. A diagram of our withdrawal detection approach is shown in Figure 7. Since we are using a one minute window, our model requires one minute

of data to be collected by a wearable biosensor before it can classify whether that person is in the withdrawal or non-withdrawal state.

List of References

[1] D. R. Wesson and W. Ling, "The clinical opiate withdrawal scale (cows)," *Journal of psychoactive drugs*, vol. 35, no. 2, pp. 253–259, 2003.

[2] J. K. Nuamah, F. Sasangohar, M. Erranguntla, and R. K. Mehta, "The past, present and future of opioid withdrawal assessment: a scoping review of scales and technologies," *BMC medical informatics and decision making*, vol. 19, no. 1, p. 113, 2019.

[3] D. Castaneda, A. Esparza, M. Ghamari, C. Soltanpur, and H. Nazeran, "A review on wearable photoplethysmography sensors and their potential future applications in health care," *International journal of biosensors & bioelectronics*, vol. 4, no. 4, p. 195, 2018.

[4] M. Kelsey, R. V. Palumbo, A. Urbaneja, M. Akcakaya, J. Huang, I. R. Kleckner, L. F. Barrett, K. S. Quigley, E. Sejdic, and M. S. Goodwin, "Artifact detection in electrodermal activity using sparse recovery," in *Compressive Sensing VI: From Diverse Modalities to Big Data Analytics*, vol. 10211. International Society for Optics and Photonics, 2017, p. 102110D.

[5] G. Vila, C. Godin, O. Sakri, E. Labyt, A. Vidal, S. Charbonnier, S. Ollander, and A. Campagne, "Real-time monitoring of passenger's psychological stress," *Future Internet*, vol. 11, no. 5, p. 102, 2019.

[6] H. Tanaka, K. D. Monahan, and D. R. Seals, "Age-predicted maximal heart rate revisited," *Journal of the american college of cardiology*, vol. 37, no. 1, pp. 153–156, 2001.

[7] A. Dehghani, O. Sarbishei, T. Glatard, and E. Shihab, "A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors," *Sensors*, vol. 19, no. 22, p. 5026, 2019.

[8] E. H. Chartoff and W. A. Carlezon Jr, "Drug withdrawal conceptualized as a stressor," *Behavioural pharmacology*, vol. 25, p. 473, 2014.

[9] Y. S. Can, N. Chalabianloo, D. Ekiz, and C. Ersoy, "Continuous stress detection using wearable sensors in real life: Algorithmic programming contest case study," *Sensors*, vol. 19, no. 8, p. 1849, 2019.

[10] H. F. Posada-Quintero and K. H. Chon, "Innovations in electrodermal activity data collection and signal processing: A systematic review," *Sensors*, vol. 20, no. 2, p. 479, 2020.

[11] F. Shaffer and J. Ginsberg, "An overview of heart rate variability metrics and norms," *Frontiers in public health*, vol. 5, p. 258, 2017.

CHAPTER 4

Model Development

In this chapter, we discuss the curation of our data set, how we will evaluate the machine learning models we develop, and the results of our model training process.

4.1 Data Curation

In this section, we will describe the data curation process. The data curation process involves extracting portions of the patient data, and creating two separate datasets for training and testing.

Given that the subjects in our dataset were examined by clinicians intermittently (every 30 minutes to an hour), we do not have the ground-truth about the patients health state at all times. Consequently, we curate the biosensor data collected from the 16 patients by only extracting patient data where we are reasonably confident of their health state (i.e., neutral, intoxicated or withdrawal). Only this curated data is used for training our detection models and evaluating their efficacy.

In total, we used data from 16 different patients in this study. Each patient had their physiological data collected anywhere from 30 minutes up to several hours. Of these 16 different patients, only 6 had withdrawal symptoms assessed by clinicians. One of the 6 patients that had data in the withdrawal state, one also had usable data assessed in the neutral state. The remaining 10 patients had data assessed in the neutral state, intoxicated state, or both.

A total of 20 minutes were extracted from the wearable biosensor data surrounding the time when the clinicians assessed the patient's state. The 20 minutes of data is comprised of the 10 minutes before and the 10 minutes after the

evaluation happened. We used these 20 minutes of data because, being in the controlled environment of the hospital, it is unlikely a patients physiological state would change drastically during this time period. However, certain parts of the 20-minute periods surrounding an assessment of a patient's state had to be excluded from our analysis due to noise issues described in Section 3.2.

The one minute segments that were assessed to be in the withdrawal state are placed into one dataset (the universal withdrawal set), and the one minute segments that were assessed to be in the intoxicated or neutral state were placed into a separate dataset (the universal non-withdrawal set) (see Figure 5). Once all patient data was placed into either the universal withdrawal or universal non-withdrawal dataset, the feature extraction process was performed.

To be able to train the classifier to detect withdrawal, we have to compensate for the idiosyncrasies in our curated dataset that originated from our data collection protocol. In order to train our classifier, we first generate the positive and negative class points (as described in Section 3.4) and then shuffle them. We then use the first 80% of the feature points for training. This allowed us to train our classifier and still have some (previously unseen by the model during training) data leftover (20%) to test its performance.

Table 4 shows the break down class points in the training and testing sets. This highlights the small sizes of both the training and test set, as well as the imbalance between the withdrawal and non-withdrawal class points.

Table 4: Breakdown of class points in training and testing sets

	Total Points	**Withdrawal Points**	**Non-Withdrawal Points**
Training Set	335	113	222
Testing Set	84	30	54

4.2 Metrics

Before we go into the details of the how the different machine learning models will be trained and tested, we provide a short overview of the metrics we use to evaluate the efficacy of our models. Since our model will classify each input example as either withdrawal (positive class) or non-withdrawal (negative class), the result from inference will fall into one of four categories. (1) True Positive (TP): A correct prediction of the positive class. (2) False Negative (FN): An incorrect prediction of the positive class. (3) True Negative (TN): A correct prediction of the negative class. (4) False Positive (FP): An incorrect prediction of the negative class. After performing inference on all input examples, we will use the number of examples in each of these four categories to calculate our models *true positive rate (TPR)*, and *false positive rate (FPR)*.

TPR is the ratio of how many positive class points were predicted correctly compared to the total number of positive class points [1]. FPR, on the other hand, is the ratio of how many negative class points were predicted incorrectly compared to the total number of negative class points [1]. The TPR and FPR for a model can be used to plot a *receiver operating curve (ROC)*. The ROC curve demonstrates how well a model classifies positive points, compared to how poor it is at classifying negative points [1]. From the ROC curve, the *area under the curve (AUC)* can be calculated to allow us to compare one machine learning model to another. The ROC AUC is the metric that we use to measure how accurately our models can identify the withdrawal class samples. The ideal ROC curve should have an AUC value that is close to 1 (perfect classification of both positive and negative class points). The ROC AUC will be the metric we will attempt to maximize during the training and testing processes.

4.3 Model Training

Once the dataset had been cleaned, the features extracted, and the training and test sets created, we can begin developing models to identify opioid withdrawal. We trained four different machine-learning classifiers to identifying opioid withdrawal. The classifiers use all 66 features extracted from the wearable biosensor data to learn what *distinguishes the withdrawal state from the non-withdrawal (neutral or intoxicated) state*. The four machine-learning (ML) algorithms developed in this training phase are: Random Forest, Decision Tree, Logistic Regression, Support Vector Machine (using a gaussian radial basis function kernel).

To first develop a baseline understanding for how our models perform during training, we performed 10-fold cross validation with our class imbalanced training dataset using all 66 features and untuned models. The ROC Curves and AUC for each algorithm for this baseline test are shown in Figure 8. What is shown in this figure is that all untuned algorithms perform fairly well using class imbalanced data, and all 66 features. The Random Forest performs especially well (AUC = 0.9690), and is still fairly close to near perfect classification. The Decision Tree (AUC = 0.8556), Logistic Regression (AUC = 0.8615), and Support Vector Machine (AUC = 0.8089) models perform well, but have much more room to improve compared to the Random Forest.

Now that we have a baseline for how these algorithms perform during cross validation without addressing the class imbalance, we decided to use two different class imbalance techniques in an effort to improve classification. These two approaches are *Synthetic Minority Oversampling Technique* (SMOTE) and *Exactly Balanced Bagging* (EBBag). The SMOTE technique samples a point p from the minority class, and randomly creates a new point that is between p and its r closest neighbors [1]. This is done until the positive class and negative class have an

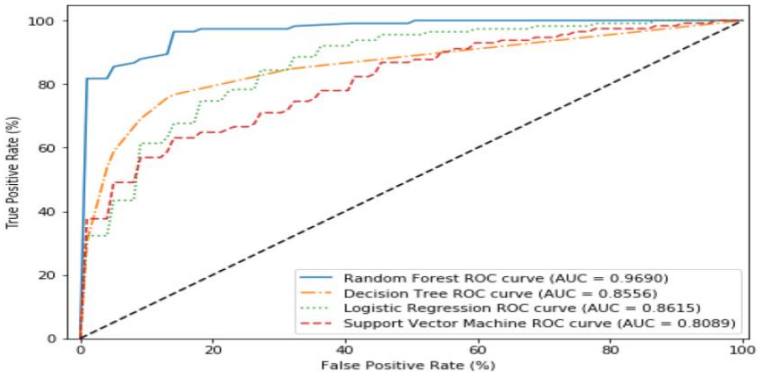

Figure 8: Average ROC curves and AUC obtained during cross validation with class imbalanced data using all features and untuned models.

equal amount of examples. This technique has been found to be very effective for handling class imbalances [1]. EBBag involves training more than one instance of a classifier on randomly under-sampled sets of the majority class which match the number of samples in the minority class [2]. This technique was done without replacement, and allows us to train two different classifiers which both used all of the withdrawal state data, and equally split the non-withdrawal state data. These two classifiers then work together to perform inference on new samples presented to them.

4.3.1 Feature Pruning

Further, in order to improve our models we *pruned* the feature set. The purpose of doing this is to find a minimal feature set which maximizes a models ability to identify the withdrawal state. The *feature pruning* process is done by starting with an empty set of features for a particular model, and at each stage adding the feature to its feature set which maximizes its ability to accurately identify withdrawal class samples during cross-validation. If all of the 66 extracted features are in the models feature set, or there are no features left which improve the models ability to accurately identify the withdrawal class samples, then the process ends. A minimal feature set is identified for each model using both of the

techniques for handling the class imbalance (SMOTE, EBBag).

4.3.2 Grid Search

Next, once we have the optimal feature set for a model, a grid search is performed to find the hyper-parameters which maximize the model's ability to accurately identify withdrawal class samples during cross-validation. Previous research has shown that using a grid search with cross validation to assess different hyper-parameter values for a model can help find which hyper-parameter values maximize a models ability to perform inference [3]. The hyper-parameters chosen to be tuned via a grid search in this study have all been used in previous research for hyper-parameter optimization [4] [5] [3]. The hyper-parameters optimized for each model are summarized in Table 5. Combined, the minimal feature set and model with tuned hyper-parameters will be used in the testing phase.

Table 5: Hyper-parameters chosen to be optimized for each model

Model	**Hyper-parameters optimized**
Random Forest	maximum depth, minimum samples per leaf, minimum samples per split, number of estimators
Decision Tree	maximum depth, minimum samples per leaf, minimum samples per split, splitting criterion
Support Vector Machine	cache size, degree
Logistic Regression	inverse regularization strength

The results of the training phase will help determine which models will be used in the testing phase for both the SMOTE and EBBag class imbalance techniques.

4.3.3 Training using SMOTE

Initially we use all 66 features to build our models using our four chosen ML algorithms. We use SMOTE to compensate for the class imbalance. Figure 9 shows the results when all features are used. We find that Random Forest (RF) performs

Table 6: Feature pruning results using SMOTE

Model	**Pruned Feature Set**
Random Forest	mean temperature, EDA mean, x-axis median, y-axis interquartile range, z-axis interquartile range
Decision Tree	mean temperature, EDA 20th percentile, EDA mean, z-axis median
Logistic Regression	mean temperature, EDA peaks, EDA mean, x-axis total power, y-axis interquartile range, y-axis zero crossing rate, y-axis minimum, y-axis skew
Support Vector Machine	mean temperature, EDA 20th percentile, y-axis median, z-axis mean

the best with a near perfect classification accuracy (AUC = 0.9785). The other algorithms do not perform as well. Next, we used the feature pruning algorithm to find the minimal feature set which maximizes classifier accuracy. Table 6 shows the reduced/pruned features list (again using SMOTE to balance the classes). Using these pruned feature sets, we find that the performance of the all of the algorithms except for the Decision Tree algorithms remain more or less the same. The Decision Tree algorithm saw improvements from an AUC of 0.8625 using all features, to an AUC of 0.9235 using the pruned feature set (see Figure 10). Finally, a grid search was performed to determine the optimal hyper-parameters for a model when using SMOTE. The model parameters which were tuned during this process using SMOTE can be seen in Table 7. The final results using SMOTE with each tuned model, and pruned feature set (see Figure 11) show the same performance as using an untuned model.

.

Table 7: Grid Search Results Using SMOTE

Model	Hyper-parameter	value
Random Forest	Criterion	gini impurity
	Minimum Samples Per Leaf	1
	Minimum Samples Per Split	2
	Number of Estimators	100
	Max Depth	15
Decision Tree	Criterion	gini impurity
	Minimum Samples Per Leaf	1
	Minimum Samples Per Split	2
	Max Depth	10
Logistic Regression	Cache Size	5
	Degree	1
Support Vector Machine	Inverse Regularization Strength	11.28

4.3.4 Training using EBBag

Initially we use all 66 features to build our models using our four chosen ML algorithms. We use EBBag to compensate for the class imbalance. Figure 12 shows the results when all features are used. We find that the Random Forest (RF) algorithm performs the best (AUC = 0.9666) of all the models when using all features. Though the Random Forest algorithm performed well using EBBag, compared to SMOTE, the AUC for all algorithms except for the Decision Tree (AUC = 0.9074) was worse than when using SMOTE. Next, we used the feature pruning algorithm to find the minimal feature set which maximizes classifier accuracy. Table 8 shows the reduced/pruned features list (once again using EBBag to balance the classes). Unlike SMOTE, using these pruned feature sets noticeably improves the performance of the four algorithms. After using the feature pruning algorithm, the Random Forest (RF) and Decision Tree (DT) algorithms performed slightly better than the results obtained using SMOTE (see Figure 13). Finally, a grid search was performed to determine the optimal hyper-parameters for a model when using EBBag. The model parameters which were tuned during this process using EBBag can be seen in Table 9. The final results using EBBag with

each tuned model, and pruned feature set (see Figure 14) show that all models performed the same or slightly better than the untuned models.

Table 8: Feature Pruning Results Using EBBag

Model	**Pruned Feature Set**
Random Forest	mean temperature, EDA mean, BVP Root Mean Square, x-axis median, y-axis minimum, BVP mean IBI
Decision Tree	mean temperature, EDA mean, x-axis median, z-axis maximum, EDA 20th percentile
Logistic Regression	mean temperature, EDA mean, EDA peaks, y-axis median, x-axis mean average derivative, x-axis total power, number of BVP peaks, z-axis zero-crossing rate, y-axis variance
Support Vector Machine	mean temperature, EDA peaks, EDA 20th percentile, y-axis maximum, z-axis median, BVP root mean square

4.3.5 Comparison Between Using Imbalanced Data And Using SMOTE or EBBag

The baseline results obtained with untuned models using imbalanced data and all 66 features (see Figure 8) can now be compared to the initial results obtained using SMOTE (see Figure 9) and EBBag (see Figure 12). When comparing the results for SMOTE with using no class imbalance technique, we can see that the results are fairly similar for all algorithms. However, when comparing the EBBag results with using no class imbalance technique, we can see that the Decision Tree algorithm performs better with EBBag (AUC = 0.9074) compared to using no class

Table 9: Grid Search Results Using EBBag

Model	Hyper-parameter	value
Random Forest	Criterion	gini impurity
	Minimum Samples Per Leaf	2
	Minimum Samples Per Split	5
	Number of Estimators	100
	Max Depth	8
Decision Tree	Criterion	entropy
	Minimum Samples Per Leaf	1
	Minimum Samples Per Split	3
	Max Depth	8
Logistic Regression	Cache Size	5
	Degree	1
Support Vector Machine	Inverse Regularization Strength	1.62

imbalance technique (AUC = 0.8556). We can also see that the Support Vector Machine model using EBBag (AUC = 0.7545) performs worse than the Support Vector Machine model using no class imbalance technique (AUC = 0.8089). In general though, our cross validation training results do not demonstrate that using either class imbalance technique (SMOTE, EBBag) provides a great advantage to simply using the imbalanced data. Since these initial results do not show that these class imbalance mitigation techniques (SMOTE, EBBag) improve classification, we decided to use the untuned models trained with imbalanced data and all 66 features, as well as the models using SMOTE and EBBag in the testing phase. We will again compare and contrast models trained using the imbalanced data with those trained using SMOTE and EBBag in the testing phase.

4.3.6 Comparison Between SMOTE and EBBag Pruned Features

During the feature pruning process for the Random Forest (RF) algorithm, three of the same features were found to maximize classification accuracy during training using both SMOTE and EBBag. These features are mean temperature, EDA mean, and x-axis median. This is a high percentage of similar features con-

sidering that the number of features in the pruned feature set for the RF algorithm using SMOTE and EBBag are five and six respectively. For all four algorithms, generally around half of the pruned feature list using SMOTE and EBBag were the same. The most commonly found features in the pruned feature sets for all of the algorithms (using either SMOTE or EBBag) are mean temperature, and EDA mean. EDA mean is one of the features highlighted in Section 3.4, which helped demonstrate the separability of the positive and negative class by plotting one feature against another.

4.3.7 Comparison Between SMOTE and EBBag Tuned Models

The tuned models using SMOTE and EBBag were similar for certain algorithms and very different for others. The Decision Tree models developed using SMOTE and EBBag were relatively similar, where only the minimum samples per split (tuned value using SMOTE was 3, and for EBBag was 2) and maximum depth (tuned value using SMOTE was 10, using EBBag was 8) were different for the tuned models. The tuned Support Vector Machine models using the two class imbalance techniques were exactly the same. On the other hand, the Logistic Regression and Random Forest tuned models were very different when using SMOTE versus EBBag. The inverse regularization strength value for the tuned Logistic Regression models using SMOTE was far higher (tuned value using SMOTE was 11.27, and for EBBag was 1.62). For the Random Forest models developed using SMOTE and EBBag the major differences were in the maximum depth (tuned value using SMOTE was 15, using EBBag was 8), minimum samples per split (tuned value using SMOTE was 2, and for EBBag was 5), and minimum samples per leaf (tuned value using SMOTE was 1, and for EBBag was 2).

Ultimately, since our training results do not show a dramatic improvement between using a tuned vs. untuned model, we decided to use both the set of

untuned models and tuned models in the testing phase.

List of References

[1] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, 2018.

[2] J. Błaszczyński, J. Stefanowski, and Ł. Idkowiak, "Extending bagging for imbalanced data," in *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*. Springer, 2013, pp. 269–278.

[3] D. Krstajic, L. J. Buturovic, D. E. Leahy, and S. Thomas, "Cross-validation pitfalls when selecting and assessing regression and classification models," *Journal of cheminformatics*, vol. 6, no. 1, pp. 1–15, 2014.

[4] T. Eitrich and B. Lang, "Efficient optimization of support vector machine learning parameters for unbalanced datasets," *Journal of computational and applied mathematics*, vol. 196, no. 2, pp. 425–436, 2006.

[5] P. Probst, A.-L. Boulesteix, and B. Bischl, "Tunability: Importance of hyperparameters of machine learning algorithms." *Journal of Machine Learning Research*, vol. 20, no. 53, pp. 1–32, 2019.

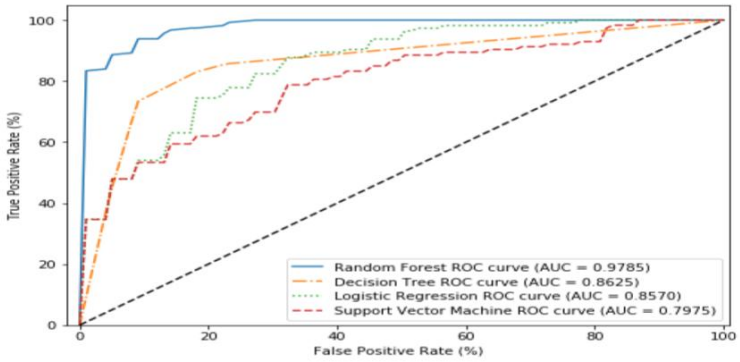

Figure 9: Average ROC curves and AUC obtained during cross validation for SMOTE using all features.

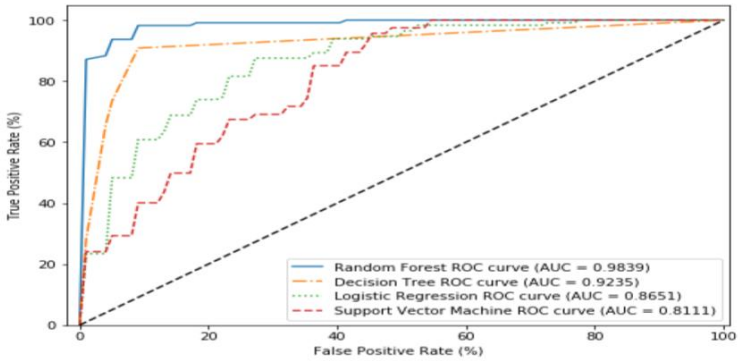

Figure 10: Average ROC curves and AUC obtained during cross validation for SMOTE using the pruned features.

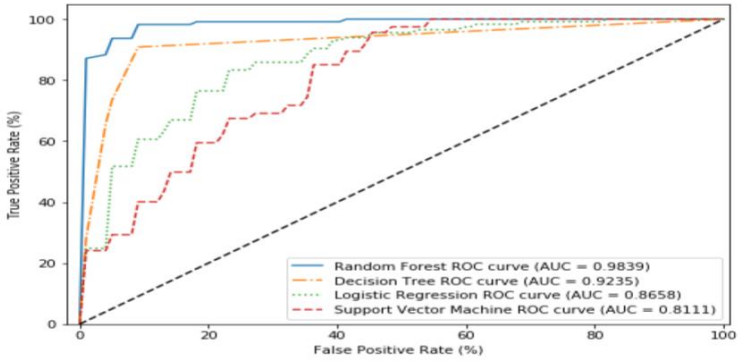

Figure 11: Average ROC curves and AUC obtained during cross validation using SMOTE for each tuned model using the pruned features.

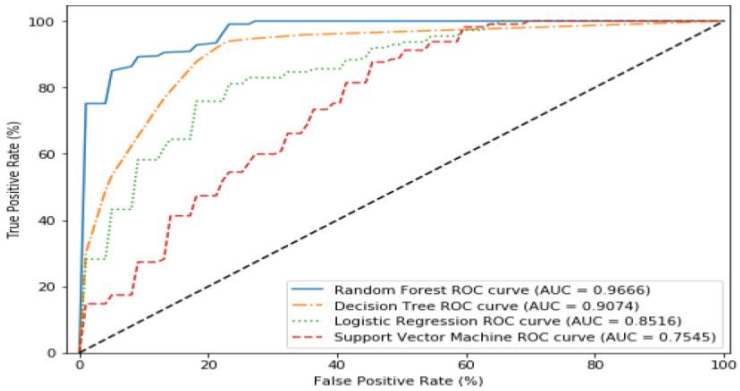

Figure 12: Average ROC curves and AUC obtained during cross validation for EBBag using all features.

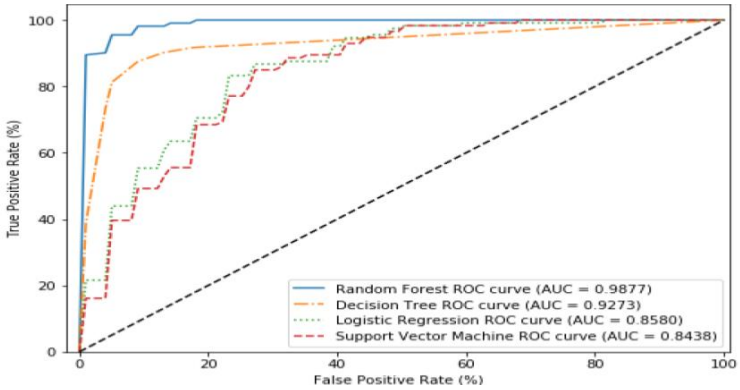

Figure 13: Average ROC curves and AUC obtained during cross validation for EBBag using the pruned features.

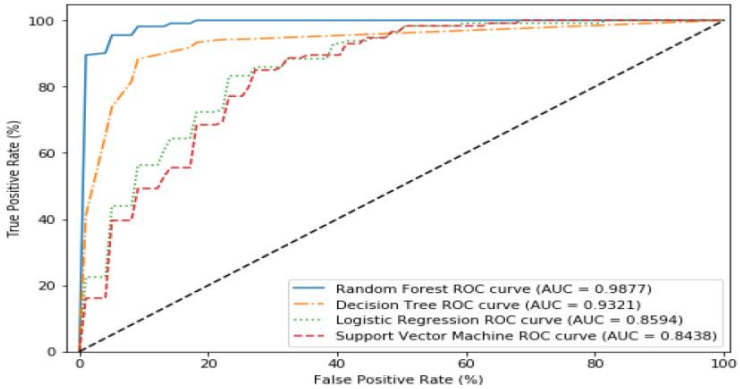

Figure 14: Average ROC curves and AUC obtained during cross validation using EBBag for each tuned model using the pruned features.

CHAPTER 5

Results

In this chapter, we discuss the results from testing our trained models on unseen data, and the limitations of our research.

5.1 Testing Results

This section describes the final results obtained in the testing phase using both the EBBag and SMOTE class imbalance methods. The results here were obtained using the test set described in Section 4.1. This test set consists of 20% of the original universal withdrawal and universal non-withdrawal datasets. The data in the test set has never been seen by any of the machine learning models during their development. The results from evaluating the trained machine learning models on the test set demonstrates how well our models generalize to unseen data.

To first develop a baseline understanding for how our models perform during testing, we performed testing with our class imbalanced testing dataset using all 66 features and untuned models. The ROC Curves and AUC for each algorithm for this baseline test are shown in Figure 15. What is shown in this figure is that all untuned algorithms perform exceptionally well using no class imbalance technique, and all 66 features. The Random Forest performs especially well (AUC = 0.9784), and again has fairly close to near perfect classification. All algorithms using no class imbalance technique, untuned models, and all 66 features performed similarly if not better on the test set compared to cross validation in the training phase.

Now that we have provided a baseline for performance We have summarized the results of the testing phase using each class imbalance technique (SMOTE, EBBag) by showing each models receiver operating characteristic (ROC) curve, and ROC area under the curve (AUC).

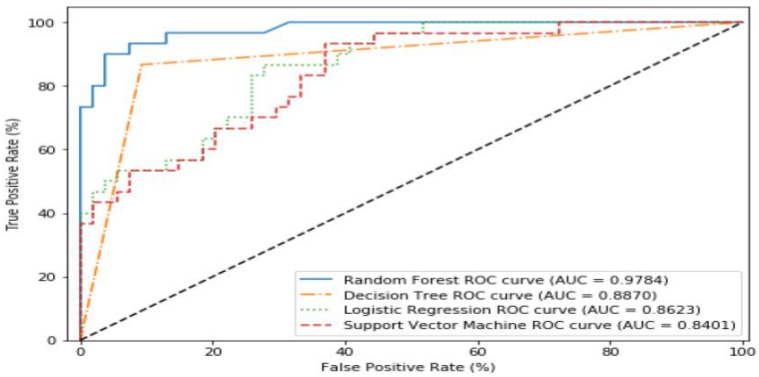

Figure 15: ROC curves and AUC obtained during testing phase using class imbalanced data with untuned models and the pruned features.

5.1.1 SMOTE Results

Figure 16 shows the ROC curve and ROC AUC obtained for each untuned model during testing using the SMOTE class imbalance technique. Figure 17 shows the ROC curve and ROC AUC obtained for each tuned model during testing using the SMOTE class imbalance technique. These results show that both the tuned and untuned models using SMOTE obtained similar results on the test set. The tuned Random Forest (AUC = 0.9880) and Decision Tree (AUC = 0.9630) models were the best two performing algorithms.

The testing results using SMOTE show for all of the algorithms except the Decision Tree, that the tuned models only performed slightly better than the untuned models. There is a noticeable improvement in classification accuracy during testing for the Decision Tree when using the tuned model (AUC = 0.9630) versus the untuned model (AUC = 0.9481). These same Decision Tree models during model development in the training procedure obtained an average ROC AUC of 0.9235, which was worse than the results obtained during testing. The Random Forest algorithm using SMOTE performed nearly equal during testing using the tuned model (ROC AUC = 0.9880) versus the untuned model (ROC AUC = 0.9873). These same models using SMOTE achieved almost the exact same results during

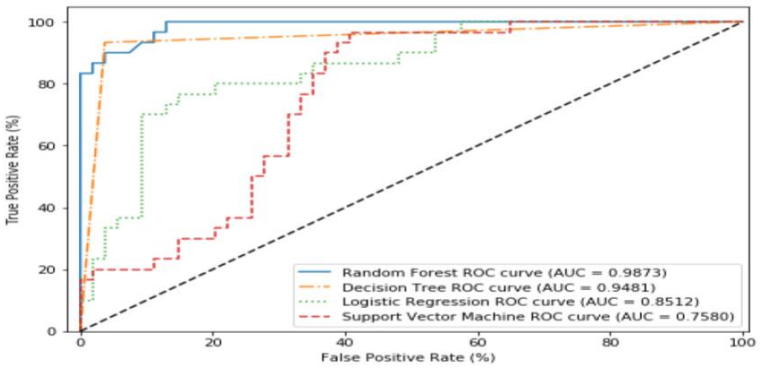

Figure 16: ROC curves and AUC obtained during testing phase using SMOTE with untuned models and the pruned features.

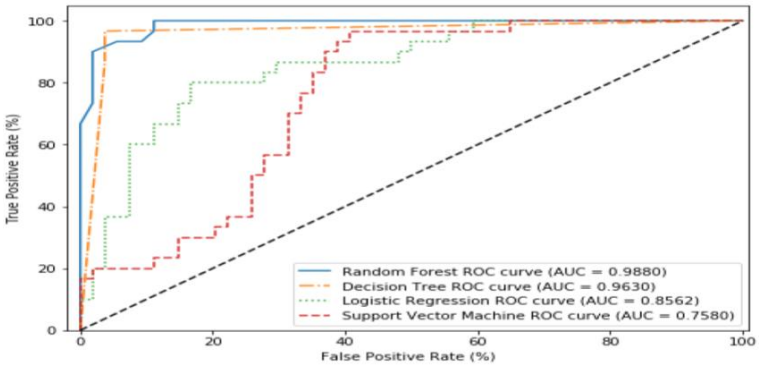

Figure 17: ROC curves and AUC obtained during testing phase using SMOTE with tuned models and the pruned features.

model development in the training procedure (both with an average ROC AUC of 0.9839). The Logistic Regression algorithm using SMOTE had very similar results during testing between the tuned model (ROC AUC = 0.8562) and the untuned model (ROC AUC = 0.8512). Again, these same Logistic Regression models using SMOTE performed only slightly better during model development in the training procedure (the tuned model obtained an average ROC AUC of 0.8658 versus the untuned model with an average ROC AUC of 0.8651). The Support Vector Machine (SVM) algorithm using SMOTE had the same results during testing using the tuned and untuned models (ROC AUC = 0.7580). These results obtained during testing with the SVM models are worse than the results obtained during

model development in the training procedure (both the tuned and untuned SVM models during training obtained an average ROC AUC of 0.8111).

5.1.2 EBBag Results

Figure 18 shows the ROC curve and ROC AUC obtained for each untuned model during testing using the EBBag class imbalance technique. Figure 19 shows the ROC curve and ROC AUC obtained for each tuned model during testing using the EBBag class imbalance technique. These results show that both the tuned and untuned models using EBBag obtained similar results on the test set. The untuned Random Forest (ROC AUC = 0.9997), and the tuned Decision Tree (ROC AUC = 0.9731) models were the best two performing algorithms.

The testing results using EBBag once again show for all of the algorithms except the Decision Tree, that the tuned models only performed slightly better than the untuned models. For the Decision Tree using EBBag, the ROC AUC using the untuned model was 0.9451 versus a ROC AUC of 0.9731 for the tuned model. These same Decision Tree models during model development in the training procedure performed worse than during testing. During model development, the tuned Decision Tree model obtained an average ROC AUC of 0.9321 and the untuned model obtained an average ROC AUC of 0.9273. The Random Forest algorithm using EBBag again achieved almost the exact same results during testing using the tuned model (ROC AUC = 0.9944) versus the untuned model (ROC AUC = 0.9997). These same models using EBBag achieved similar results during training (both untuned and tuned Random Forest models obtained an average ROC AUC of 0.9877). The Logistic Regression algorithm using EBBag also had very similar results during testing between the tuned (ROC AUC = 0.8438) and the untuned model (ROC AUC = 0.8457). These same Logistic Regression models using EBBag performed only slightly better during the training procedure (the tuned

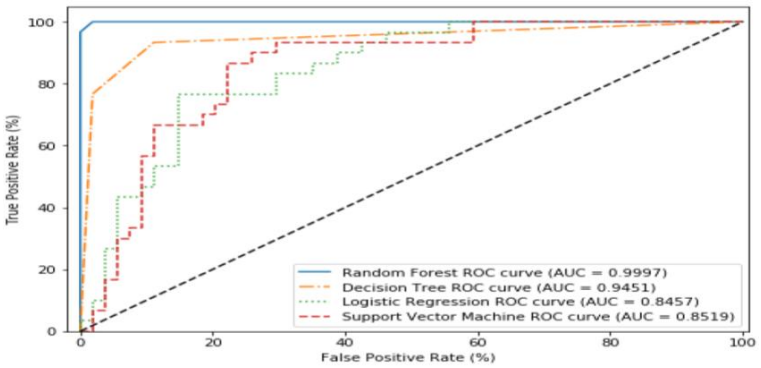

Figure 18: ROC curves and AUC obtained during testing phase using EBBag with untuned models and the pruned features.

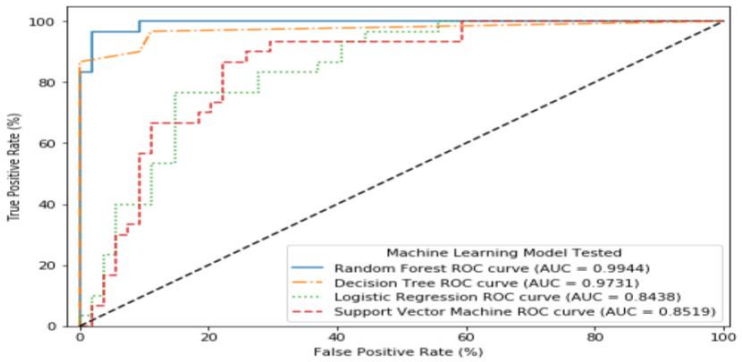

Figure 19: ROC curves and AUC obtained during testing phase using EBBag with tuned models and the pruned features.

model obtained an average ROC AUC of 0.8594 versus the untuned model with an average ROC AUC of 0.8580). The Support Vector Machine (SVM) algorithm using EBBag had the same results during testing using the tuned and untuned models (ROC AUC = 0.8519). These results obtained during testing with the SVM models are in this case slightly better than those obtained during model development in the training procedure (both the tuned and untuned SVM models during training obtained an average ROC AUC of 0.8438).

5.1.3 Comparison Between Using Imbalanced Data And Using SMOTE or EBBag

The baseline testing results obtained with untuned models using imbalanced data and all 66 features (see Figure 15) can now be compared to the results obtained using SMOTE (see Figure 16, 17) and EBBag (see Figure 18, 19). When comparing the untuned models using imbalanced data to the models using SMOTE and EBBag, we can see that the Decision Tree performs far better using SMOTE (AUC = 0.9630) or EBBag (AUC = 0.9731) compared to using an untuned Decision Tree with imbalanced data (AUC = 0.8870). The Random Forest model on the other hand only performed slightly better using SMOTE (AUC = 0.9880) or EBBag (AUC = 0.9997) compared to using an untuned Random Forest model with imbalanced data (AUC = 0.9784). It is important to note that the models using SMOTE or EBBag may or may not be tuned models, and they are using pruned feature sets found during the training phase. Surprisingly, the untuned Logistic Regression and Support Vector Machine models using imbalanced data performed similarly, if not better than those same models using SMOTE or EBBag. In general, the untuned models using imbalanced data performed well on the test set and were very capable of accurately identifying opioid withdrawal.

5.1.4 Final Takeaways

The results obtained during the training procedure using both SMOTE and EBBag found that there was little improvement in classification accuracy for all four tuned models versus untuned models. This is generally true for the result found during the testing procedure, however, it was found during testing that the tuned Decision Tree model using SMOTE and EBBag had a noticeable improvement in classification accuracy compared to the untuned model.

Further, we find that our testing results show that for some algorithms (Ran-

dom Forest, Decision Tree) there is some benefit to using SMOTE and EBBag (along with our training methodology) compared to training untuned models with imbalanced data and all 66 features. However, without using SMOTE or EBBag the baseline results using untuned models with imbalanced data and all 66 features still perform well. This shows that for this dataset these class imbalance techniques (or others) are not necessary to develop a model that can identify withdrawal with near perfect accuracy. We know this because the untuned Random Forest trained with imbalanced data and all 66 features obtained a testing ROC AUC of 0.9784.

Although the results obtained in testing are very good, they need to be understood in context. Given that this is a new area of research and the general dearth of datasets for this work, we have had to work with a small dataset. We have thus demonstrated through our training and testing process that using machine learning classifiers for detecting opioid withdrawal (in near real-time; 1 minute) from wearable biosensor data is viable. Given that this the first work in this area, we believe this is an important contribution. That being said, we do not claim to have produced generalizable classifiers for this particular task.

To further question and verify our testing results, we decided to perform leave-one-out cross validation. This testing methodology and the results obtained are discussed in detail in the next section.

5.2 Verification Of Testing Results

In this section, we describe how we attempted to verify our testing results shown in the previous section.

5.2.1 Leave-One-Out Cross Validation

In order to verify our testing results, we decided to perform *leave-one-out cross validation*. Performing *leave-one-out cross validation* helps us verify that our

models are not just simply learning what withdrawal looks like for each of the individual six subjects with withdrawal data in the training set. All of the models used in this process were untuned, and all 66 features were used. To perform *leave-one-out cross validation*, we performed 6-fold cross validation where for each fold we leave out the entire set of withdrawal data for a particular subject to be used in the test set. Then we randomly sample 10% of the non-withdrawal data to be in the test set as well. For each fold, we train our four different algorithms using all of the withdrawal data for 5 patients, as well as the remaining 90% of the non-withdrawal data. We then test those models using the test set that consists of all of the withdrawal data for one patients, as well as the randomly sampled 10% of the non-withdrawal data. We then average the results for all six folds to obtain an average accuracy for each model.

Leave-one-out cross validation was performed using the imbalanced data, SMOTE, and EBBag. Again, all of the models trained were untuned, and the training and testing data consisted of all 66 features. In Figure 20, we show the average ROC curves and AUC for each algorithm using the imbalanced dataset. In Figure 21, we show the average ROC curves and AUC for each algorithm using SMOTE. In Figure 22, we show the average ROC curves and AUC for each algorithm using EBBag.

What the leave-one-out cross validation results show is that on average the Random Forest algorithm still performs well, especially using SMOTE (avg. AUC = 0.8338). The Logistic Regression and Support Vector Machine algorithms on the other hand struggle to classify nearly as well as they did during testing, with their average ROC AUC falling between 0.6411 and 0.7134. The Decision Tree algorithm struggles when using imbalanced data (AUC = 0.6771), but performs much better using SMOTE (AUC = 0.7430) and EBBag (AUC = 0.7908). Overall, these results

are worse than those obtained using the 20% test set, but they demonstrate that the models still have some predictive power for identifying withdrawal in unseen subject data.

5.3 Limitations

The results of this study show that there is promise in using wearable biosensors to identify opioid withdrawal. However, there are two major limitations in our work that need to be addressed in future work.

The first major limitation of this study is the small amount of opioid withdrawal state data that we had access to. This is in large part due to that fact that collecting this type of data is subject to if and when an OUD patient experiences opioid withdrawal symptoms.

For the data that was able to be collected from OUD patients experiencing withdrawal in this study, there were issues with noise in the data that may have rendered portions of it to be unusable in this analysis.

The second limitation of this study is that we only collected data that represents *naloxone induced* opioid withdrawal. The symptoms of a precipitated withdrawal onset by naloxone will be shorter, and possibly more extreme than the spontaneous withdrawal that occurs when opioid use is reduced or stopped altogether [1]. Therefore, these results may not generalize well to detecting spontaneous opioid withdrawal.

List of References

[1] H. D. Kleber, "Pharmacologic treatments for opioid dependence: detoxification and maintenance options," *Dialogues in clinical neuroscience*, vol. 9, no. 4, p. 455, 2007.

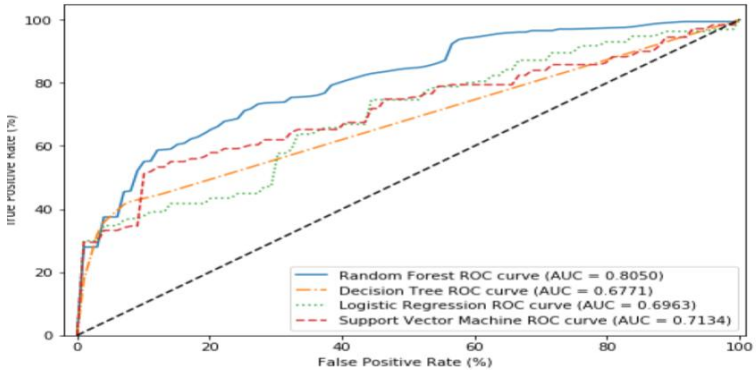

Figure 20: Average ROC curves and AUC obtained during *leave-one-out cross validation* using **imbalanced data**.

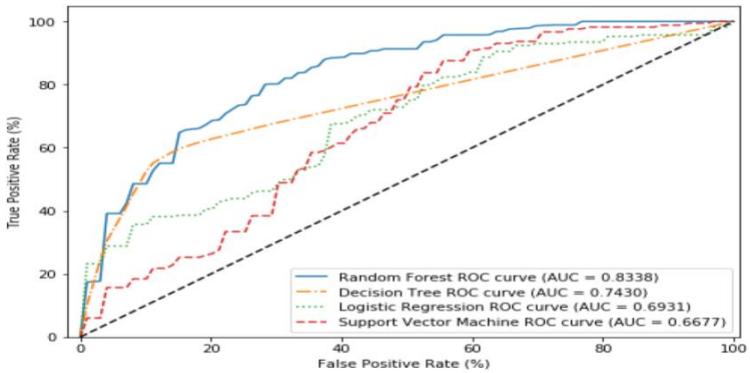

Figure 21: Average ROC curves and AUC obtained during *leave-one-out cross validation* using **SMOTE**.

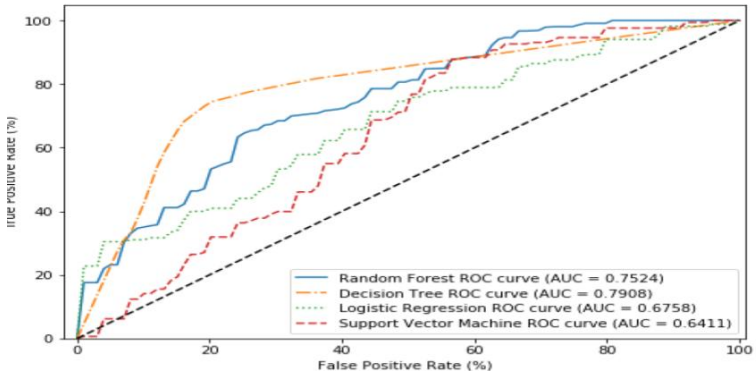

Figure 22: Average ROC curves and AUC obtained during *leave-one-out cross validation* using **EBBag**.

CHAPTER 6

Conclusion And Future Work

In this chapter, we draw conclusions from the results shown in Chapter 5, and discuss how we plan to continue our work.

6.1 Conclusions

In this study, we proposed a method for developing a set of machine learning models to identify opioid withdrawal using data collected from a wearable biosensor. We found through our training and testing procedures that the Random Forest model produced the best results. The test accuracy using this model was nearly perfect (ROC AUC=0.9997). This model only used the pruned feature set developed during the model training process, and did not undergo any model tuning with a grid search.

Further, we verified our testing results by performing leave-one-out cross validation, where for each fold we left out the entire set of withdrawal data for a particular subject to be used in the test set. Then for each fold we trained our four different algorithms using all of the withdrawal data for the remaining 5 subjects, and then tested these models with the holdout set that contained all of the withdrawal data for one particular subject. Averaging across all 6 folds, the top performing model achieved an average ROC AUC of 0.8338 (untuned model, using SMOTE and all 66 features). This supplementary test helped verify that during the training and testing phases our models are not just simply memorizing what withdrawal looks like for the six different subjects with withdrawal data in the training set. Instead, there are clear patterns in the withdrawal data that our models are able to pick up on in order to accuracy identify opioid withdrawal using data collected with a wearable biosensors.

The results during training and testing did reveal that class imbalance techniques such as SMOTE or EBBag are not necessary to develop a model for this dataset that can identify withdrawal with near perfect accuracy. During testing, the untuned Logistic Regression and Support Vector Machine models using imbalanced data and all 66 features performed equal or better than those same models using SMOTE or EBBag. On the other hand, our testing results showed that the Decision Tree and Random Forest models using SMOTE and EBBag outperformed the untuned Random Forest and Decision Tree models using imbalanced data and all 66 features. Again, it is important to note that the models using SMOTE or EBBag may or may not be tuned models, and they are using pruned feature sets found during the training phase. Our methodology has shown that performing feature pruning helps improve model performance, and this provides evidence for why the models using SMOTE or EBBag may perform better than untuned models using imbalanced data and all 66 features.

Our testing results also showed that the grid search process performed during model development for our four algorithms did not prove to be very beneficial. This will be discussed further in Section 6.2.

Overall, the results obtained in this study need to be understood in context. This research has only demonstrated the viability of the proposed methodology, and it has not produced a perfect classifier for this particular task.

The most important outcome from our analysis is that identifying opioid withdrawal using data collected from wearable biosensors can be successfully done. This provides a justification for further research, and to explore the use of wearable biosensors to monitor OUD patients during their detoxification process. In the future, health care providers will be able to equip an OUD patient undergoing the detoxification process with a wearable biosensor. During the detoxification pro-

cess, this wearable biosensor will collect and transmit an OUD patient's biometric data to a cloud-based server, where an opioid withdrawal identification algorithm (similar to ours) will process and classify their data as being in *withdrawal or non-withdrawal*. If a patient's biometric data is found to be in the withdrawal state, their health care provider will be alerted, and they can then *implement a personalized treatment plan for mitigating the opioid withdrawal symptoms*. Implementing personalized treatments will help stop or lessen their patient's withdrawal symptoms, and as a result will help prevent the relapse of opioid use and overdose.

6.2 Future Work

In the future, we plan to improve upon this work in three different ways. (1) We plan to collect additional data from OUD patients experiencing naloxone induced withdrawal symptoms. The increase in data will help to address the issue of model generalizability. (2) Similarly, data will also be collected from OUD patients experiencing spontaneous withdrawal symptoms onset from discontinuing or limiting their opioid use. This data would allow us to understand how well a model built to detect naloxone induced withdrawal symptoms can generalize to spontaneous withdrawal symptoms. It would also enable us to study building a model to identify spontaneous withdrawal symptoms, or both precipitated and spontaneous withdrawal symptoms. (3) We will further investigate how performing hyper-parameter tuning can improve our models performance. Our current approach of using a grid search was not significant enough to add any sizable improvement to model performance. In future work we may instead decide to try performing a randomized grid search prior to choosing a set of parameters for a grid search, or try a different hyper-parameter tuning approach altogether.

BIBLIOGRAPHY

"CDC's efforts to prevent opioid overdoses and other opioid-related harms," Nov 2019. [Online]. Available: https://www.cdc.gov/opioids/framework/index.html

Bailey, G. L., Herman, D. S., and Stein, M. D., "Perceived relapse risk and desire for medication assisted treatment among persons seeking inpatient opiate detoxification," *Journal of substance abuse treatment*, vol. 45, no. 3, pp. 302–305, 2013.

Błaszczyński, J., Stefanowski, J., and Idkowiak, Ł., "Extending bagging for imbalanced data," in *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*. Springer, 2013, pp. 269–278.

Can, Y. S., Chalabianloo, N., Ekiz, D., and Ersoy, C., "Continuous stress detection using wearable sensors in real life: Algorithmic programming contest case study," *Sensors*, vol. 19, no. 8, p. 1849, 2019.

Carreiro, S., Chintha, K. K., Shrestha, S., Chapman, B., Smelson, D., and Indic, P., "Wearable sensor-based detection of stress and craving in patients during treatment for substance use disorder: A mixed methods pilot study," *Drug and Alcohol Dependence*, p. 107929, 2020.

Carreiro, S., Smelson, D., Ranney, M., Horvath, K. J., Picard, R. W., Boudreaux, E. D., Hayes, R., and Boyer, E. W., "Real-time mobile detection of drug use with wearable biosensors: a pilot study," *Journal of Medical Toxicology*, vol. 11, no. 1, pp. 73–79, 2015.

Carreiro, S., Wittbold, K., Indic, P., Fang, H., Zhang, J., and Boyer, E. W., "Wearable biosensors to detect physiologic change during opioid use," *Journal of medical toxicology*, vol. 12, no. 3, pp. 255–262, 2016.

Castaneda, D., Esparza, A., Ghamari, M., Soltanpur, C., and Nazeran, H., "A review on wearable photoplethysmography sensors and their potential future applications in health care," *International journal of biosensors & bioelectronics*, vol. 4, no. 4, p. 195, 2018.

Chalana, H., Kundal, T., Gupta, V., and Malhari, A. S., "Predictors of relapse after inpatient opioid detoxification during 1-year follow-up," *Journal of addiction*, vol. 2016, 2016.

Chartoff, E. H. and Carlezon Jr, W. A., "Drug withdrawal conceptualized as a stressor," *Behavioural pharmacology*, vol. 25, p. 473, 2014.

Chintha, K. K., Indic, P., Chapman, B., Boyer, E. W., and Carreiro, S., "Wearable biosensors to evaluate recurrent opioid toxicity after naloxone administration: a hilbert transform approach," in *Proceedings of the... Annual Hawaii International Conference on System Sciences. Annual Hawaii International Conference on System Sciences*, vol. 2018. NIH Public Access, 2018, p. 3247.

Dehghani, A., Sarbishei, O., Glatard, T., and Shihab, E., "A quantitative comparison of overlapping and non-overlapping sliding windows for human activity recognition using inertial sensors," *Sensors*, vol. 19, no. 22, p. 5026, 2019.

Eitrich, T. and Lang, B., "Efficient optimization of support vector machine learning parameters for unbalanced datasets," *Journal of computational and applied mathematics*, vol. 196, no. 2, pp. 425–436, 2006.

Guk, K., Han, G., Lim, J., Jeong, K., Kang, T., Lim, E.-K., and Jung, J., "Evolution of wearable devices with real-time disease monitoring for personalized healthcare," *Nanomaterials*, vol. 9, no. 6, p. 813, 2019.

Kelsey, M., Palumbo, R. V., Urbaneja, A., Akcakaya, M., Huang, J., Kleckner, I. R., Barrett, L. F., Quigley, K. S., Sejdic, E., and Goodwin, M. S., "Artifact detection in electrodermal activity using sparse recovery," in *Compressive Sensing VI: From Diverse Modalities to Big Data Analytics*, vol. 10211. International Society for Optics and Photonics, 2017, p. 102110D.

Kleber, H. D., "Pharmacologic treatments for opioid dependence: detoxification and maintenance options," *Dialogues in clinical neuroscience*, vol. 9, no. 4, p. 455, 2007.

Krstajic, D., Buturovic, L. J., Leahy, D. E., and Thomas, S., "Cross-validation pitfalls when selecting and assessing regression and classification models," *Journal of cheminformatics*, vol. 6, no. 1, pp. 1–15, 2014.

Mahmud, M. S., Fang, H., Wang, H., Carreiro, S., and Boyer, E., "Automatic detection of opioid intake using wearable biosensor," in *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018, pp. 784–788.

Nuamah, J. K., Sasangohar, F., Erranguntla, M., and Mehta, R. K., "The past, present and future of opioid withdrawal assessment: a scoping review of scales and technologies," *BMC medical informatics and decision making*, vol. 19, no. 1, p. 113, 2019.

Posada-Quintero, H. F. and Chon, K. H., "Innovations in electrodermal activity data collection and signal processing: A systematic review," *Sensors*, vol. 20, no. 2, p. 479, 2020.

Probst, P., Boulesteix, A.-L., and Bischl, B., "Tunability: Importance of hyperparameters of machine learning algorithms." *Journal of Machine Learning Research*, vol. 20, no. 53, pp. 1–32, 2019.

Shaffer, F. and Ginsberg, J., "An overview of heart rate variability metrics and norms," *Frontiers in public health*, vol. 5, p. 258, 2017.

Singh, R., Lewis, B., Chapman, B., Carreiro, S., and Venkatasubramanian, K., "A machine learning-based approach for collaborative non-adherence detection during opioid abuse surveillance using a wearable biosensor," in *Biomedical engineering systems and technologies, international joint conference, BIOSTEC... revised selected papers. BIOSTEC (Conference)*, vol. 5. NIH Public Access, 2019, p. 310.

Tanaka, H., Monahan, K. D., and Seals, D. R., "Age-predicted maximal heart rate revisited," *Journal of the american college of cardiology*, vol. 37, no. 1, pp. 153–156, 2001.

Tharwat, A., "Classification assessment methods," *Applied Computing and Informatics*, 2018.

Vila, G., Godin, C., Sakri, O., Labyt, E., Vidal, A., Charbonnier, S., Ollander, S., and Campagne, A., "Real-time monitoring of passenger's psychological stress," *Future Internet*, vol. 11, no. 5, p. 102, 2019.

Wesson, D. R. and Ling, W., "The clinical opiate withdrawal scale (cows)," *Journal of psychoactive drugs*, vol. 35, no. 2, pp. 253–259, 2003.